Vol.1 No.8 Nov. 1982

SYNCHRO SETTE

THE SUBSCRIPTION MAGAZINE FOR
THE T/S-1000 and THE ZX-81 MICROCOMPUTERS

SYNCHRO-SETTE

is published monthly by:

The S&S Company
388 West Lake Street
Addison, Illinois 60101

(312) 628-8955

HISTORY

THE FIRST LADY OF COMPUTERS

What did the poet, Lord Byron have in common with the computers we have today?

Probably nothing personally, but he was the father of Lady Augusta Ada Byron Lovelace, who became a pioneer in early computer technology.

The story goes that Lord Byron left his wife and daughter when Augusta was just an infant. In the tradition of society of that period of history, she was strictly raised by her mother and educated at home in the classical subjects of that time.

Augusta demonstrated a great affinity for mathmatics and while still a teenager, married Lord King in 1835 who later became the Earl of Lovelace. He helped her in her interest in mathematics and through him, was introduced to Charles Babbage.

Now, Charles Babbage had invented a calculating machine that is considered by most as being the first device that had computing capabilities and was christened The Analytical Engine. This machine could compute equations to 20 decimal places and worked on a system of rods and levers.

Lady Lovelace became enthralled with the device, and in 1842 she translated a paper about the engine from French to English and added many of her own thoughts to the translation. She was the first person to describe terminology such as Subroutines and Loops - concepts accepted today in almost every computer language and yet about one hundred years before their implimentation.

Mr. Babbage and Lady Lovelace became close friends and partners for many years in their joint effort with the Analytical Engine.

Lady Lovelace never received true recognition for her efforts because of the anti-feminist attitudes of that time. She was also a compulsive gambler on horse races. History does not record whether she used the Analytical Engine to predict the outcome of any of those races.

The story ends sadly as Lady Lovelace developed cancer and died at the age of 36.

EDITOR RAMBLINGS

GROWTH OF SYNCHRO-SETTE

We, like many supporting companies for the Sinclair computers, are suffering from growing pains. We have increased our staff and still are having problems meeting deadlines. The most obvious result of this is how late our issues keep coming out. Our goal is to have all of our customers receive their current issues by the first week of the month. The old adage, THE HURRIEDER I GO, THE BEHINDER I GET has never been more applicable. I used to think that it wasn't much more work to prepare a magazine for 1000 subscribers than it was for 100. Boy was I wrong. The phones ring constantly and most calls are long distance. People are asking us for information in every possible area besides our own products. The most common inquiry regards printers.

We feel since these people are investing in a long distance call, we should try to give them as much information as possible. We also receive many inquiries by mail, currently over 4000 per month and growing all the time. This means that the staff and myself cannot dedicate the time each day that is necessary to meet our deadlines and commitments. I, myself am spending on the average between 12 to 14 hours per day, six days a week trying to keep pace. We are trying desperately to get the next issue out so that you will receive the magazine and cassette before the holidays. Please bear with us for the late issues because we feel that we are finding ways and means to improve this situation.

AMATEUR RADIO AND
THE SINCLAIR COMPUTER

QZX is a journal covering amateur radio and the Sinclair family of computers, e.g. ZX-80, Micro-Ace, ZX-81 and the T/S-1000 and is designed for owners of these computers who are licensed radio amateurs.

The subscribers to this publication talk to the entire world using their computers to perform some facet of the communication.

The newsletter is a continuation of the one that was started by Marty Irons, was published 4 times a year and was sent to about 250 subscribers. The new management estimates present membership at over 300.

To find out more information about this exciting new field of communications, contact:

QXZ
2025 O'Donnell Drive
Las Cruces, N. Mexico, 88001

Ambrose 'Bo' Barry, W4GHV/5
Editor

Alex F. Burr, K5xy
Publisher

COMPUTERS IN SCHOOLS

Many schools are wondering what computers are the best for teaching first timers. The T/S-1000 and ZX-81 is usually at the top of the list because of price. The drawbacks, however, fall mainly into two catagories:

A - The Sinclair machines are not taken seriously because of their size and the membrane keyboard. The competition has propagated the myth that they are just toys and the users will soon outgrow their capabilities.

B - They do not have many of the frills their more expensive counterparts have, such as color, high resolution graphics and sound, all of which are important to the development of user's computer literacy.

Bunk - nothing could be further from the truth! The size and membrane keyboard is actually an advantage in that the computer is less susceptible to damage from key contacts being exposed to the atomosphere, dirt and spillage such as in the non-membrane competitors. If damage does occur, the repair or replacement bill will be much smaller than for the competitor's models.

The frills such as sound, color and high-resolution graphics have little place in the real world, except in arcade games and maybe fancy business graphs.

This little computer is actually a very powerful and dynamite little machine.

But putting all that aside, one area that has been almost completely overlooked is the teaching of computer science to that age group that should be the youngest to be exposed to it, mainly the 5th thru 8th grades of grammer school.

This group has been demonstrated to have problems with certain motor responses, that is, hand to eye co-ordination using a typewriter keyboard more so than their older peers. It has been shown that when groups of the same age level were given the use of Sinclair computers versus other more expensive and elaborate computers with all the bells and whistles, the Sinclair group of users learned faster and retained more EVEN IF THEY HAD TAKEN A PREVIOUS CLASS WITH THE OTHER COMPUTERS.

Why? What was this magic that Uncle Clive was able to weave?

Think of it logically. What is the one thing that the T/S-1000 and the ZX-81 have,that the competition doesn't?

Single key-stroke command input! That's right. Put yourself in the place of a 10 or 11 year old. You probably never touched a typewriter before for any serious reason and now you are placed in front of a computer and expected to type in programs. You are constantly reading a list of program lines, hunting for the proper keys to press and then checking the screen to see if the input was correct. Heaven help you if it didn't. You would have to backspace, again the eyes constantly flitting between the screen and the keyboard while the fingers are trying to do the right thing.

To put in a command such as PRINT requires 5 keystrokes. The same with the command INPUT. On the Sinclair machines it only requires 1 keystroke, not just for these commands but all commands, functions and logical operators. The end result is that the youngster has less to do and is rewarded sooner for his or her efforts. This means that impressions are made in the child's mind within a shorter time from the initiation of the effort to the keyboard and the attention span

time required is shorter.

The child retains more of what's happening.

Isn't this our goal, whether we be a responsible parent or teacher? Would not the same principles apply to the teaching of adults?

When Radio Shack first introduced the TRS-80, it had a short-hand BASIC language that abbreviated almost all of the commands. It was a simple language to understand and could accomplish many things.

They later introduced their Level II BASIC conversion kit, which promised to turn that computer into a very powerful machine - which it did. This language even today in my opinion is still the most powerful BASIC language there is.

The only problem was that the difference between the two languages was like night and day. It was like graduating from grammer school and going directly into a university.

The manual that came with the kit wasn't much help to the uninitiated, either. There was no index. Some people even made money simply by writing an index for the manual and selling it through classified ads.

The manual was termed a User's Manual. This meant that users were expected to read it cover to cover and correlate the information to previously learned languages from other computers.

The buying public that converted their machines was for the most part unaware of what they were in store for. Many of them wanted the old ROMs put back in the computers or simply sold them.

Many, though, were irreversibly bitten by the computer bug and trudged on, forcing themselves to learn the new language through trial and error, not to mention excessive phone calls to the more learned and/or purchase of anything they could get their hands on that would help their cause.

In those days there was very little help. Thank God those days are behind us now. Books, publications, places to call, etc. all exist now, at least more so than they did then, and we're only talking 6 years ago. The Sinclair computer's BASIC language is very similiar to the TRS-80's.

I have used and taught programming classes on almost all the popular computers until 1981 and I firmly believe the quickest way to achieve computer literacy is on a machine that has single keystroke entry of all the multi-character commands, functions and logical operaters and the only computers that I am aware of that have these features are the ones from Uncle Clive.

WATCHMAKER'S GUIDEBOOK

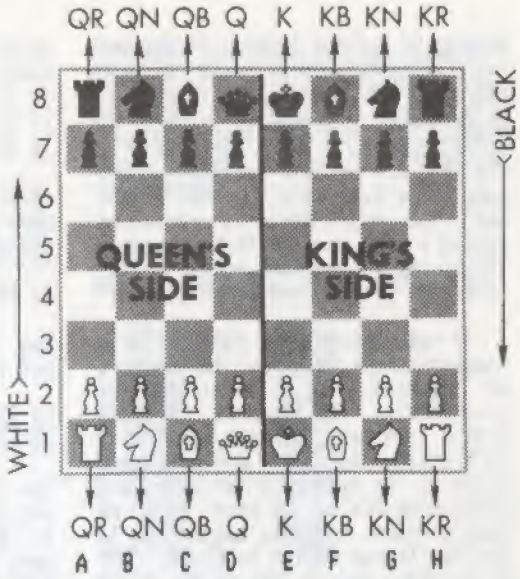
TSG Enterprises has just released a resource book for the T/S-1000 computer. The book is called "The Watchmaker's Guidebook to the TIMEX/SINCLAIR Computers" and is available by mail-order from:

TSG Enterprises, Guidebook
54 Richwood Place
Denville, NJ, 07834

for 3.95 plus 1.00 postage & handling. The book is 44 pages and contains a directory of suppliers with a brief description of their wares. About 120 software suppliers, 50 hardware suppliers and 20 ancillary suppliers are included. In addition, the Guidebook contains directories of user groups, of TIMEX/SINCLAIR specific magazines and books and an index to articles appearing in general personal computer magazines about the TIMEX/SINCLAIR computer. Also included in the Guidebook is an introductory chapter about writing a program in BASIC.

(CONT. PAGE 15)

CHESS FOR THE SINCLAIR



CHESS FOR THE SINCLAIR COMPUTER

Many people purchase home computers for the prime purpose of entertainment. The thought of having a game playing partner that not only provides a constant challenge but is available at the convenience of the user is a most attractive combination.

Chess, through the years, has been a game popular with people in all walks of life. Computer chess is not only attractive for these reasons, but it usually offers different levels of challenge and also presents the unique capability of not gloating when it is victorious (Boris chess computers non-withstanding).

The first commercially available chess computer was marketed by Fidelity Electronics in Chicago in 1976 for $175. It was called the CHESS CHALLENGER and played awful chess. But what made it unique is that it could play chess and always made legal moves.

After some months, the CHESS CHALLENGER 3 was introduced and owners of the original model could have theirs upgraded to perform like the new model which had 3 levels of difficulty. It wasn't long before other companies started marketing their models and an entirely new market proliferated.

The first version on tape that I am aware of which was available for desk top computers commercially was MICRO-CHESS, which came out for the TRS-80 and later the PET and APPLE. None of these programs represented a serious challenge until SARGON-2 (on tape) and the CHESS CHALLENGER-10 (dedicated chess playing computer) became available in late 1977.

These programs were unique in that they could see many plys (half moves) ahead compared to their predecessors and had routines that could sacrifice and play for position. They quickly became champions in all the microcomputer tournaments and even fared well by defeating some of the mainframe computer programs in the large computer chess tournaments.

Since that time, these programs have been improved to the point where there is a commercially marketed chess computer from Fidelity Electronics (now in Florida) that uses a new version of the Sargon programs that plays chess at a U.S. Chess Federation level of over 1800 in 2 to 3 minute moves. This is good enough to beat over 99% of the chess playing public the majority of the time.

There are a few chess programs

available for the Sinclair computer too. Gladstone Electonics, (1-800) 833-8400, sells three of them of which the best is called ZX CHESS II CHESS MASTER. This program is claimed to be the strongest chess game to be available on the ZX-81 and has many features including saving a game on tape to be resumed later, printing the moves on the screen or printer and board set-up.

In case you are not familiar with computer chess notation, the board's 64 squares are identified by co-ordinates. The board is lettered from A to H left to right and from 1 to 8 bottom to top. White always plays at the bottom of the board and black from the top. A move such as King's pawn from the second row to the fourth row woulds be E2-E4. The E is for the letter of the fifth column where the King normally resides.

These statements regarding the ZX Master program seemed to me to be pretty strong claims, so I decided to have a tournament between this program and SARGON 2 on the TRS-80. I set the levels on each computer to make each move in an average time between 2 to 3 minutes. I did not keep a record of the time each program took for any of the moves or the games but I had them play 2 games against each other, each playing white once and black once and the following are the results with my annotations:

Game #1
White - ZX Master, Black - Sargon 2

1	E2-E4	E7-E6
2	D2-D4	D7-D5
3	B1-D2	G8-F6
4	E4-E5	F6-G8
5	G1-F3	B8-C6
6	F1-D3	F8-E7
7	0-0	C8-D7
8	D1-E2	A8-B8
9	D2-B3	C6-B4
10	C1-D2	D7-C6
11	F1-E1	B7-B5
12	D2xB4	E7xB4
13	C2-C3	B4-E7
14	B3-A5	D8-D7
15	A5-B3 ?	A7-A6
16	A1-C1	B8-D8
17	C1-C2	G8-H6
18	A2-A3	0-0
19	C2-D2	H6-G4
20	D2-C2 ?	F8-E8 ?

20 F7-F6 seems to have more merit in an attempt to remove the oppresive pawn at E5.

21	F3-D2	G4-H6

Too many wasted moves on both sides and it doesn't stop!

22	D2-F3	H6-G4
23	F3-D2	G4-E5 ??

The first big mistake by Sargon. I was all set to reset both computers to higher time levels because of move repetition when this move was made. Suprisingly, Sargon played a pretty stong game and actually had a fair attack going at a later point in this game and then it made its second big mistake.

24	E2xE5	E7-F6

Try as you may to find one but black has no redeeming attack!

25	E5-F4	D7-D6
26	F4-E3	E6-E5

Applause, Applause!

27	E3-G3	E5-E4
28	G3xD6	D8xD6
29	D3-F1	C6-D7
30	E1-E3	F6-G5
31	E3-E1	

Best move available for white. Staying on the 3rd rank loses the rook because of - 31 E3-G3 G5-F4.

31		F7-F5
32	B3-C5	G8-F7?

Both of these programs seem to prefer using the King as an attacking piece. I would prefer pushing some of the back-row pawns.

33	C5-B7	D6-G6
34	D2-B3	G5-F4
35	B7-C5	E8-E7

Both programs don't seem to mind trading pieces that could become essential in an attack.

36 C5xD7	E7xD7

Of course, 2 rooks can do more damage than a bishop and a rook. The potential attack looks very strong but it never materializes and not because of superior play by White.

37 B3-C5	D7-D6
38 A3-A4	B5xA4
39 C5xA4	A6-A5

Black should have concentrated on doubling up the rooks!

40 C3-C4	G6-H6
41 G2-G3	H6-G6
42 F1-H3	F7-F6
43 H3-G2	D5-C4

The second idiot move of the game by Black. The attack it had going was admirable considering it was down a piece. This move has absolutely no redeeming value. The potential for the King-side attack is gone unless White makes a stupendous blunder which it doesn't.

44 G3xF4	D6xD4
45 E1-C1	F6-E7
46 C2xC4	

Crumbs!

	D4xC4
47 C1xC4	G6-G4
48 C4xC7 ch	

And White even gets back the pawn!

	E7-D6
49 C7-A7	G4xF4

More crumbs!

50 A7xA5

Black is no match for White's superior forces and goes on to be mated on the 81st move in a rather dull end-game.

I was very disappointed in Sargon's play in this game. I have found that at this level, it could solve any Mate-In-Two problem that I gave it in less than 3 minutes and over 75% of them in less than 2 minutes. I was sure that the rematch would redeem Sargon and restore my confidence.

Game #2
White - Sargon 2, Black - ZX Master

1	C2-C4	C7-C6
2	G1-F3	G8-F6
3	B1-A3	E7-E6
4	B2-B3	D7-D5
5	E2-E3	F8-C5
6	C1-B2	B8-A6
7	F1-E2	D8-E7
8	D1-C1	0-0
9	0-0	C8-D7
10	D2-D4	C5xA3

The first strong move by Sargon, attempting to break up Black's board center control and establish its own.

11 B2xA3	A6-B4

Good counterplay!

12 A3xB4	E7xB4
13 F1-D1	D5xC4
14 C1xC4	B4xC4

How these programs like to trade, makes no difference whether they are ahead or behind.

15 B3xC4	A8-D8
16 A1-B1	D7-C8

Black's B7-B5 makes more sense with the potential to liberate the bishop.

17 F3-E5	F6-E4
18 D1-D3	F8-D8
19 E2-F3	E4-G5
20 F3-H5	E8-E7
21 B1-D1	A7-A6

The plan behind this last move escapes me.

22 A2-A3

As does this one.

(CONT. PAGE 17)

STRING MANIPULATION

Good afternoon class, I see you're all looking fit today! Today's subject will be on string manipulation.

Can anyone give me an example of string manipulation? Yes, the gentleman in the rear has his hand up.

No, that's wrong - and please put that yo-yo away. That's not the kind of string we're discussing here today.

No, the strings I'm interested in are string variables in BASIC computer programs, such as the BULLETIN BOARD programs and the HOME BUDGET programs. Let me ask another question. What would be the minimum amount of program lines needed to allow a MARQUEE type message scroll across the screen on the Sinclair computer?

One person says 20 - another 15 - still another 13.

Would you believe it could be done in five? That's right, five. ENTER the following program into your computer:

SCROLLING DEMO PROGRAM

```
10 INPUT A$
20 LET A$ = " 32 blank spaces "
   + A$ + " "
30 PRINT AT 10, 0; A$( TO 32 )
40 LET A$ = A$(2 TO LEN A$) +
   A$(1)
50 GOTO 30
```

RUN the program and ENTER any message you want or as much as your computer will hold. Then press the ENTER key. The appearance of movement of the message occurs, but in actuality, a string of 32 characters is constantly being thrown on the screen, changed, thrown again on the screen, changed, etc.

The message is not really moving but only gives the illusion of movement. We can get the same illusion in the opposite direction by changing line 40 to read:

```
40 LET A$ = A$(LEN A$) + A$( TO
   LEN A$ -1)
```

Instead of RUNning the program again and retyping in the message, just ENTER the command - GOTO 30. Of course, we are not limited to just alpha-numeric character entry. We can also introduce graphic characters to produce some pretty startling effects. The beginning of last month's cassette program "BULLETIN" used two of these routines to intoduce the Bulletin Board.

Pretty nifty, huh? This is one method we can use to manipulate string variables.

There are others, though, such as word processing. Let us say that we had a word processor such as the 1K "MICROPRO" on last month's cassette. This program created one variable that could be increased or decreased in length.

But, what if we wanted to insert a word into that string variable? Enter the following simple program:

WORD PROCESSING DEMO PROGRAM

```
10 LET B$ = ""
20 LET C$ = "    "
100 LET A$ = INKEY$
120 IF CODE A$ = 114 THEN
    GOSUB 300
130 IF CODE A$ = 115 THEN
    GOSUB 400
140 IF CODE A$ = 23 THEN
    LET A$ = " "
150 IF CODE A$ = 220 THEN
    GOSUB 500
190 IF CODE A$)63 THEN
    LET A$ = ""
200 LET B$ = B$ + A$
210 PRINT AT 11,0 ; B$; AT 12,
    LEN B$; CHR$ 136; AT 12,
    LEN B$; " "; AT 11,LEN B$; C$
220 GOTO 100
300 LET C$ = B$(LEN B$) + C$
310 LET B$ = B$(TO LEN B$ -1)
320 LET A$ =""
330 RETURN
400 LET B$ = B$ + C$(1)
410 LET C$ = C$(2 TO LEN C$)
420 LET A$ = ""
430 RETURN
500 LET C$ = C$(2 TO LEN C$)
510 LET A$ = ""
520 RETURN
```

RUN this program and enter the following sample text one character at a time.

"HELLO, HOW ARE YOU?"

This program will only allow one line of text to be entered on the screen. The blinking cursor appears under the next character position to be inputted.

CURSOR LEFT & RIGHT - The SHIFT arrows allow the cursor to move left or right through the text. You will notice that if you hold any character key down, the character will repeat itself.

DELETE - To delete a character, press the SHIFT key and the (0) key.

BLANK SPACE - To put a blank space in, press the SHIFT key and the asterisk (*) key. CAUTION - Do not press BREAK or SHIFT RIGHT ARROW for a blank space.

INSERTION - To insert text into existing text, just place the cursor under the position of the text where you want the text to be inserted. An example would be under the (A) in (ARE) and type in "THE HECK ". See how easy it is?

This is all done by manipulating the string variables, (B$) and (C$) with the (TO) function. For you more experienced programmers it should be a good exercise to try to combine these techniques with some of the other word-processing routines and programs from our past issues to produce a word-processor that can hold many lines of text.

As you can see from these examples, the (TO) function is an extremely powerful string manipulating device. Not only is it powerful but also extremely fast, even in the SLOW mode. Most word processors have to be written in machine language in order to get the speed necessary to make them workable. It is my belief that because of the speed and power of the (TO) function, a very fast word processing program could be written in BASIC that would have most if not all of the features of established programs for other microcomputers, such as the ELECTRIC PENCIL and SCRIPTSIT programs. These other programs can sell for $100 and more.

But be that as it may, let us move on to other uses for string manipulation.

DATA FORMATTING

In past issues and from cassette programs, we have learned that if we wanted to create a list of data, such as a customer file, we would have to have separate variable arrays for every item category to be inputted. The man in the rear has his hand up again. Yes sir?

He wants to know what a variable array is. Good question.

ARRAYS

An array is a word used to format a category of variable information. Let us say that we took our little address book that we all have either on our person or at home and we wanted to put all those names, addresses, phone numbers and other related information into the computer's memory. Let us say that we never expected the list to exceed 100 people. In the FILE program on the OCT/82 cassette, we find this is accomplished by dimensionalizing out a string variable for each item to be entered using the format DIM N$(100,n) where (N$) would be the string variable assigned to the particular item entered, the (100) is the maximum amount of those items that will be entered and (n) would be a number that would represent the maximum amount of characters in that item. We now have the format set for the creation of an array to allow the input of a maximum of 100 items. Let us say that this array is for last names.

But, what if we needed more than one set of items, such as another array the input of a maximum of 100 first names and other separate arrays for addresses, phone numbers, cities or towns, zip codes, etc. We would need to dimensionalize more string variables, one for each category of items that we have and using up one letter of the alphabet for each category.

Can anyone tell us what the limit would be to the amount of arrays we can have in a program? Yes, the gentleman in the rear has his hand up again?

He says that after the 26 letters of the alphabet are used up, no more arrays can be created.

Once again, your answer is wrong, sir, but I can easily understand how you came to that conclusion. There is another method which allows you as many variables as the computer's memory will allow, but first let me discuss some of the methods used on some of the more expensive computers with disk drives.

SEQUENTIAL VS. RANDOM ACCESS DATA STORAGE

Data storage and retrieval from disk can be done with two different methods. Sequential storage and retrieval requires all data to be in the computer's memory banks as it is being processed. This means that in a name and address program, all the name and address information is in the computer and is available for recall from the computer's memory chips.

Random Access data storage works on an entirely different principal. The total data is never entirely in the computer's memory banks - only the data which is pertinent to the operation of that part of the program. The program allows the data to be inputted and then stores it on the disk. The program then forgets the data and moves onto the next input. When the data is to be retrieved, it is displayed on the CRT or printed onto paper and then forgotten. Sometimes routines in the program exist to total numeric values.

The advantage of Random Access storage is that the disk storage media can hold a lot more data than the computer's memory can, the advantage being maybe 10, 20 or 100 times as much. The disadvantage is that program totaling routines require the disc drive to always be running so that it is searching for the required data and is therefore more time-consuming. This is particularly true for sort routines that require disk variable swapping.

You are probably wondering what any of this has to do with more than 26 variable arrays being available for the Sinclair computer. The reason I brought all this up is because of the methods used for Random Access storage where a whole bunch of string variables can be stored in one main string variable.

FIELDS, ELEMENTS AND SUB-VARIABLES

ENTER the following program:

```
10 CLS
20 IF LEN A$)9 THEN GOTO 5
30 PRINT ,, A$
40 FOR N = 1 TO LEN A$
50 PRINT N;
60 NEXT N
70 PRINT "FIRST NUMBER?"
80 INPUT A
90 IF A<1 OR A) LEN A$ THEN
   GOTO 80
100 PRINT ,,,, "LAST NUMBER?"
110 INPUT B
120 IF B< LEN A$ - A THEN
    GOTO 110
130 PRINT ,,,, A$(A TO B)
140 GOTO 5
```

RUN this program and ENTER a string of characters no more than 9 characters in length. After you press the ENTER key, you will see a number under each character corresponding with that character's position in the string. The program will now allow you to break out a segment of that string from the first number entered to the last number entered. The program will recycle so that you can try it over again.

The gentleman in the rear has another question. Yes, sir?

He asks if this isn't the exact opposite of what we set out to accomplish? He says that he can see how a string of characters can be broken apart but is wondering how sets of strings of characters can be brought together into one common formatted variable?

Patience, my friend. We must crawl before we walk and roll before we crawl. ENTER the following program:

FIELD PROGRAM

```
10 DIM A$(5,32)
20 FOR N = 1 TO 5
30 SCROLL
40 PRINT "NAME?",
50 INPUT B$
60 LET A$(N, TO 12) = B$
70 PRINT B$
80 SCROLL
90 PRINT "ADDRESS?",
100 INPUT B$
110 LET A$(N,13 TO 24) = B$
120 PRINT B$
130 SCROLL
140 PRINT "PHONE NO.?",
150 INPUT B$
160 LET A$(N,25 TO 32) = B$
170 PRINT B$
180 SCROLL
190 SCROLL
200 NEXT N
210 SCROLL
220 SCROLL
230 PRINT "PRESS ENTER TO
    SEE VARIABLES :::"
240 SCROLL
250 SCROLL
260 SCROLL
270 INPUT B$
280 FOR N = 1 TO 5
290 SCROLL
300 PRINT A$(N)
310 SCROLL
320 PRINT "123456789012345678901
    23456789012"
330 SCROLL
340 SCROLL
350 NEXT N
360 SCROLL
370 SCROLL
380 PRINT "PRESS ENTER TO SEE
    SEGMENTS :::"
390 INPUT B$
400 SCROLL
410 SCROLL
420 FOR N = 1 TO 5
430 PRINT "NAME",A$(N, TO 12)
440 SCROLL
450 PRINT "ADDRESS",A$(N, 13
    TO 24)
460 SCROLL
470 PRINT "PHONE NO.",A$(N,
    25 TO 32)
480 SCROLL
490 SCROLL
500 NEXT N
```

RUN this program and enter five names, addresses and phone numbers. You are limited to 12 characters for the name, 12 characters for the address and 8 characters for the phone numbers. Purposely enter more than this limit for some of the items. You will notice later that the computer will automatically truncate them to the limited amount of characters.

After the fifth set is entered, you will be given the instruction:

PRESS ENTER TO
SEE VARIABLES :::

Press ENTER and the variables will be displayed with the 32 numbers below them for identifying the character positions. The numbers start with 1 and go to 9 which is followed by 0, 1, 2 etc. The first character of the 2 digit numbers is omitted.

This entire program was written with only two string variables and yet we have three separate pieces of information under program control. Actually, these three pieces of information are in only one string variable.

CREATING DATA FIELDS

I previously mentioned Random Access data manipulation methods. These methods incorporate a technique to create FIELDS within a string variable. If you look at your screen, you will see the data you previously entered now formatted into fields within single string variables withing an array. The array is the 5 total string variables. The fields of each variable are as follows:

- character positions 1 through 12 represent the NAME information.

- character positions 13 through 24 represent the ADDRESS information.

- character positions 25 through 32 represent the PHONE NUMBER information.

FIELDING RULES

The following rules apply when data is entered into these fields:

- the first character of the FIELD item or element corresponds with the first FIELD position assigned to that element in the dimensionalized string variable (line 10) by the formatting program routine assigned to that task (lines 60, 110 and 160).

- all characters following the first character of the FIELD element will take the next available positions in the corresponding dimentionalized string variable.

- all positions of the dimensionalized string variable are null spaces (blank) until they are replaced with characters (line 10 makes all the positions blank and lines 60, 110 and 160 replaces these blank spaces with the data derived from lines 50, 100 and 150).

- if the amount of characters of the data to be inserted into a FIELD position exceeds the amount of space allowed, the data will be truncated (chopped off) from its right side to accommodate the space allowed.

- if the amount of characters of the data to be inserted into a FIELD position is LESS than the amount of space allowed, blank spaces will follow the data to take up the remaining space.

Study these variables and the positions of the characters in them. You see the message at the bottom of the screen:

PRESS ENTER TO SEE SEGMENTS :::

Do so and you will see all of the elements are broken out of the string variables. We now have demonstrated one method of putting a bunch of elements into a single string variable and breaking them out again.

But in the immortal words of that famous and world reknowned philosopher, Alfred E. Neuman, "You Ain't Seen Nuttin' Yet".

SORTING BY FIELDS

What you are about to see is an extremely powerful string manipulating technique. Add the following routine to the last program. CAUTION - DO NOT RUN THE PROGRAM AFTER THIS ROUTINE IS ENTERED!

```
510 SCROLL
520 SCROLL
```

```
530 PRINT "PRESS ENTER
    TO CONTINUE :::"
540 INPUT B$
600 FAST
610 CLS
620 SLOW
630 PRINT "WHICH FIELD DO YOU
    WANT TO SORT","(NAME -1/
    ADDRESS -2/ PHONE -3)"
640 INPUT A
650 IF CODE STR$ A<29 OR CODE
    STR$ A>31 THEN GOTO 640
660 FAST
670 GOTO 680 + (A*100)
700 LET Y = 1
710 LET Z = 12
720 GOTO 1000
800 LET Y = 13
810 LET Z = 24
820 GOTO 1000
900 LET Y = 25
910 LET Z = 32
1000 CLS
1010 LET N = 5
1020 LET N = INT N/2
1030 IF N = 0 THEN GOTO 1200
1040 LET J = 1
1050 LET K = 5 - N
1060 LET I = J
1070 LET L = I + N
1080 IF A$(I,Y TO Z)<=A$(L,
     Y TO Z) THEN GOTO 1150
1090 LET B$ = A$(I)
1100 LET A$(I) = A$(L)
1110 LET A$(L) = B$
1120 LET I = I - N
1130 IF I<1 THEN GOTO 1150
1140 GOTO 1070
1150 LET J = J + 1
1160 IF J>K THEN GOTO 1020
1170 GOTO 1060
1200 SLOW
1210 GOTO 210
```

You will notice that the above routine is almost identical to the SHELL SORT on page 11 of the OCT/82 issue. Lines 1010 and 1050 have changed the variable (B) to (5). This variable and number correspond with the total amount of strings to be sorted.

Now ENTER (GOTO 210). Press ENTER and the data you entered previously should be displayed. Press ENTER again and the FIELD ELEMENTS will be displayed. Now press ENTER and choose one of the fields you want sorted. The following 2 displays will show the data being sorted by the command you gave. If you chose NAME - 1, the names will be in alphabetical order. If address or phone number were chosen, the string variables would have been sorted by those fields.

I might mention that I am not aware of any other computer that allows this format and speed to accomplish this task and a program could be written to allow as many ELEMENT FIELDS per string variable as the computer's memory would allow.

This could be a very valuble technique in writing an inventory program and is used in the BILLPAYER MACHINE of the HOME BUDGET package marketed by SYNCHRO-SETTE.

This, by no means, is all that can be done by manipulating string variables. The techniques are only limited by the programmer's imagination.

For your homework, try to write a program that has at least 12 elements per variable, 50 total variables and allows a sort for any element category.

HEY, YOU IN THE BACK! I TOLD YOU TO PUT THAT YO-YO AWAY!

(ED. RAMB. CONT.)

INDIANA USERS' GROUP

Robert DeLisle has started a Sinclair computer club. For further information contact:

CCC Users' Group
The Calumet Computer Club
1588 Beverly St.
Hammond, IN, 46324

HOME BUDGET REVIEW

The following is a review of two home budget programs, Package #4/16K - 14.95 from Synchro-Sette:

THE CHECKBOOK MACHINE

This program is suitable for applications from home to medium size businesses. It is self running and when loaded, the screen displays the following menu:

1 - TO ENTER TRANSACTIONS
2 - TO CREATE EXPENSE CATEGORIES
3 - TO SEE EXPENSE CATEGORIES
4 - TO LIST TRANSACTIONS
5 - BANK RECONCILIATION
6 - TO SAVE DATA ON TAPE
7 - TO CLEAR TRANSACTIONS
8 - TO EDIT TRANSACTIONS

Extensive documentation is included that covers almost every conceivable checkbook situation.

To build a monthly data file, the user first establishes the EXPENSE CATEGORIES with prompt (2). As many as 40 categories such as Food, Clothing, Rent, Utilities, Auto, Insurance, etc. can be labelled. The categories can always be reviewed at any time with prompt (3).

After the categories are created, prompt (1) allows transactions to be entered. The program will ask for the first check number and will automatically number the checks from that point on. Transaction entries are coded for checks, deposits and bank service charges. If mistakes are made, breakout to the main menu is possible and prompt (8) allows editing.

Prompt (4) allows the user use of three routines:

1 - TO SEE TRANSACTIONS
2 - TO SEE EXPENSE TOTALS
3 - TO SEE TRANSACTION TOTALS

Routine (1) scrolls all the transactions as entered. Routine (2) shows only the categories that debits were written into and displays their totals. Routine (3) shows the totals amounts for checks, deposits and bank service charges.

Prompt (5) from the main menu allows the user to balance the checkbook against the bank's statement. A nice feature is the entry method for outstanding checks. Only the check numbers need be entered. The program will automatically cross-reference them for the amounts against the ones in memory. If the final total does not equal the bank's statement total, the most probable areas for error will be listed.

Prompt (7) allows the user to start a new month's file by clearing out all the transactions but LEAVING THE CATEGORIES INTACT. This simplifies entry for the next month

as the user goes directly to entering the transactions.

Prompt (8) allows the user to edit any mistakes made in any transaction entry and prompt (6) allows any file to be saved on tape, whether it be a partial or full month of entries.

THE BILLPAYER MACHINE

Simplicity itself to use, this program allows the user to organize unpaid bills in a data file that can be constantly updated as old bills are paid and new ones are entered.

Self running and menu driven, the user needs only three items of information to be entered:

1 - the date the bill is due
2 - the creditor's name
3 - the amount due

The user sits down at the computer with all of the bills in front of him or her and starts entering the data. The program will then sort the data and provide a list with the oldest bill at the top and the newest at the bottom.

As bills are paid, they can be deleted or edited. If a bill is a onetime debt, such as a magazine subscription, once paid it can be deleted completely. If it is a regular bill with regular payment amounts, the user can edit only the date. Bills with date and amount changes, such as utilities, can have just those areas edited.

The sort routine is almost instantaneous, and of course the data can be stored on tape.

		E7-C7
23	H5-G4	B7-B5
24	D1-C1	F7-F6
25	E5-F3	G5-E4
26	C4xB5	A6xB5
27	F3-D2	E4xD2
28	D3xD2	D8-D6
29	C1-C5	D6-D5
30	D2-C2	

Hurrah! Let Black initiate the trade and White still has the same blocade with the chance to win a pawn.

		D5xC5
31	C2xC5	G8-F7
32	G4-F3	

Wrong! G4-E2 would win either the knight's pawn or the bishop's pawn because after Black's C8-A6, A3-A4 wins the pawn because of a double pin. White ignores this potential on its next move, also.

		C8-B7
33	G1-F1	F7-E7
34	F1-E2	E7-D6
35	E2-D3	E6-E5
36	F3-E4	G7-G5
37	G2-G3	C7-E7
38	F2-F4	G5xF4
39	E3xF4	E5xD4
40	C5-F5	C6-C5

Not the obvious. Superior move by black to keep the pawn concentration on the Queen side.

41	E4xB7	E7xB7
42	F5xF6ch	D6-D5
43	F6-F5ch	D5-D6
44	F5-F6ch	

The game is a draw. I set the levels of both machines to higher difficulties and the repetition went on for 6 more moves. Either side has the potential to Queen pawns and a simple mistake could be disaster for either White or Black.

A half point for each side. Final score, ZX Master 1 1/2 points, Sargon 2 - 1/2 point.

ZX Master is definitely one of the stronger programs. Next month ZX Master will be pitted against a stronger program than Sargon 2, Z-Chess 3.

Dear Ed,

I had a little trouble with the "INFLATION" - 1K program as printed in the September issue of SYNCHRO-SETTE.

In order to get it to display as shown on page 7, I changed the program as follows:

```
178 PRINT ;; I$;" THAT COST ";C;
    TAB 31;" IN 1982, WILL COST "
    ;F;TAB 31;"IN ";Y;"."
```

The program now runs without any problems. I am a new subscriber - would you please send info on the Mindware plain-paper printer?

J.F. Lapunzina - Brooklyn, NY

Dear Joe,

Thanks for the correction. We don't know if Mindware will ever have their printer available through mail-order. They have told us that there is an undeterminable delay due to not being able to be okayed by the FCC.

Rumor has it that Mindware is getting out of the mail-order business anyway and investigating selling their products through retail outlets.

The specs on the 119.95 MW-188 printer are:

- uses 1 3/4" adding machine paper - uses a standard typewriter ribbon cartridge - impact, not thermal - buss connector sandwiches between computer & RAMpack - prints 16 characters per line.

The last spec is interesting in that if a screen or program listing is to be printed from a 32 character format, the left side can be printed and then the right side, and the two pasted together.

Ed.

Dear Ed,

I cannot load the OCT82 tape sent. I have had trouble in the past with the MRX3 oxide tape when I wished to load my peograms. My guess is that the tape head on my machine and this tape do not agree.

Please reconsider the manufacturer of the tapes you use for duplication. Most tapes have dropouts. Perhaps Maxell UDXL II does not because of stringent quality control.

M. Herscovitch - Montpelier, VT

Dear Monty,

Of all the tapes we tested, the

MRX3 by Memorex had the best quality control standards. We have had less than one half of one percent defective tapes. There are other tapes that are of even better quality but the price would be prohibitive unless the subscriber would wish to pay some dollars extra for each tape sent.

Cost and stringent quality control aside, many people have found that less expensive tapes outperform the so called top-of-the-line tapes almost every time. This is not a random phenomenon.

The computer, by design, requires input of high frquency sound pulses. Lower frequency pulses interfere with the computer's capability to accept the high pulses by distorting them.

Top of the line tapes are receptive to sound reproduction at all frequency levels and as a result can accept mid-range and low level sounds at a higher volume level than some of the cheaper tapes, even thought the high frequency sound reproduction of the top line tape may be superior.

The MRX3 tape has the unique characteristic of having exceptional high frequency absorbption with very little mid-range to low frequency absorbption. True, drop-outs can occur. This is usually because of lack of the metallic oxide in the tape itself but can be caused by other factors, such as the recording equipment itself, particularly when the heads pick up a magnetic field. We are constantly de-magnetizing the recording heads of our equipment.

The tapes we used previously had an almost 30% defect rate and we checked each one individually. This may be why they seemed to be superior to some subscribers compared to the MRX3. It is now impossible to do this kind of checking because of our subscription rate.

Ed.

Dear Ed,

Is your name really Ed? Did the Old Professor ever moonlight as a bartenter in Walla Walla, Washington? He looks like the guy that slipped me a mickey there one night when I was shanghied to the Orient.

Seriously though, I bought my TS-1000 last month and have returned it to be traded two times because of the same problem. The screen would all of a sudden white out and the keyboard was unresponsive. It seems improbable that three machines would all have the same problem. Is there something I could be doing wrong or does Timex have a problem with quality control?

Tex Shula - Dallas, TX

Dear Tex,

Of course my name isn't Ed. Ed is short for editor. I come from Loisiana but who wants to be called Louise?

I asked the old Professor if he was ever in Washington and he says he was indeed and remembers you that night at the bar. He was paid enough money for slipping you the mickey to buy his first computer and that's how he got started. He told me to thank you for helping his new career begin.

The quality control problem you mention is probably no more severe than with any other manufacturer. I have heard stories about quality control problems with every popular micro. To have three bad ones in a row, however, is more than one would or should expect. From what you told me, unless your house electrical service continuously experiences power surges, I don't think it's anything that could be caused by the user. It could have been just a bad batch of computers from the store you got them from.

Ed.

SHORT PROGRAMS

2K minimum

How about a program to get you into the spirit for the holidays? This little ditty should be a suprise to all but the most advanced programmers.

HAPPY HOLIDAYS

```
10 FOR N = 1 TO 176
20 PRINT CHR$ 128 + CHR$ 128 +
   CHR$ 128 + CHR$ 128;
30 NEXTN
40 FOR N = 1 TO 10
50 LET A = (15*RND) + 2
60 LET B = (25*RND) + 2
70 PRINT AT A,B; CHR$ 1 + CHR$
   132; AT A + 1, B - 1; CHR$ 1
   + " " + CHR$ 132; AT A + 2,
   B - 2; CHR$ 1 + "   " +
   CHR$ 132; AT A + 3, B; CHR$ 5

80 NEXT N
900 LET A$ = ""
910 FOR N = 1 TO 30
920 LET A$ = A$ + CHR$ 128
930 NEXT N
940 LET B$ = "17316618118119012817 3
    1801771741691661901841281711831
    8017812818419017916817318318015
    8184170185185170128"
950 FOR N = 1 TO LEN B$ - 2 STEP 3
960 LET A$ = A$ + CHR$ VAL B$(N TO
    N + 2)
970 NEXT N
1000 PRINT AT 11,1; A$(TO 30)
1010 LET A$ = A$(2 TO LEN A$) +
    A$(1)
1020 UNPLOT RND*62, RND*43
1030 GOTO 1000
```

After you have impressed your friends with that one, you may want to show them something that is close to being magic. Did you know that the computer can figure out what day of the week any date in history falls on? You input the date, month and year (numbers only) and this program will do the rest. The only catch is, the date must be after the year 1752. I doubt if anyone out there has a birthday before that.

DATES, PAST, PRESENT & FUTURE

```
10 DIM D$(7,9)
20 LET D$(1) = "SUNDAY"
30 LET D$(2) = "MONDAY"
40 LET D$(3) = "TUESDAY"
50 LET D$(4) = "WEDNESDAY"
60 LET D$(5) = "THURSDAY"
70 LET D$(6) = "FRIDAY"
80 LET D$(7) = "SATURDAY"
90 PRINT ,,,,"ENTER THE YEAR?"
100 INPUT Y
110 IF Y < 1753 THEN GOTO 100
120 PRINT "ENTER THE MONTH?"
130 INPUT M
140 IF M < 1 OR M > 12 THEN
    GOTO 130
150 PRINT "ENTER THE DATE?"
160 INPUT D
170 IF D < 0 OR D > 31 THEN
    GOTO 160
180 LET K = INT (.6 + (1/M))
190 LET L = Y - K
200 LET O = M + 12*K
210 LET P = L/100
220 LET Z1 = INT (P/4)
230 LET Z2 = INT P
240 LET Z3 = INT ((5*L)/4)
250 LET Z4 = INT (13*(O + 1)/5)
260 LET Z = Z4 + Z3 - Z2 + Z1
    + D - 1
270 LET Z = Z - (7*INT (Z/7)) +1
280 LET Y$ = STR$ Y
290 PRINT ,,M; "/";D; "/"; Y$
    (LEN Y$ - 1 TO LEN Y$);
    " FALLS ON A "; D$(Z)
300 INPUT A$
310 CLS
320 RUN
```

It is not that hard to use this routine to put a calender on the screen for any month.